บทที่ 2

เทคโนโลยีเอเจนท์แบบเคลื่อนที่

หลังจากที่มีการนำเสนอระบบเอเจนท์แบบเคลื่อนที่ในปีค.ศ. 1994 คือ ระบบ Telescript (White 1996) แล้วพบว่ามีการวิจัยที่เกี่ยวข้องกับเอเจนท์แบบเคลื่อนที่เกิดขึ้นอย่างกว้างขวาง มีการกำหนดมาตรฐานของระบบเอเจนท์แบบเคลื่อนที่เพื่อให้ระบบที่มาจากต่างผู้ผลิตกันสามารถทำงานร่วมกันได้ รวมทั้งมีการพัฒนาระบบเอเจนท์แบบเคลื่อนที่ทั้งในเชิงวิจัยและเชิงพาณิชย์เป็นจำนวนมาก นอกจากนี้ยังมีการนำเอเจนท์แบบเคลื่อนที่ไปประยุกต์ใช้กับงานในด้านต่าง ๆ เช่น การพาณิชย์อิเล็กทรอนิกส์ (Dasgupta et. al 1999) การสืบค้นข้อมูลบนระบบฐานข้อมูลแบบกระจาย (Papastavrou et. al 2000) และการเฝ้าติดตามระบบเครือข่ายเพื่อจัดการเครือข่ายแบบกระจาย (Du et. al 2003) เป็นต้น

เนื้อหาในบทนี้จะกล่าวถึง คำนิยามที่เกี่ยวข้อง แนวคิดเรื่องเอเจนท์แบบเคลื่อนที่ภาษาในการพัฒนาระบบเอเจนท์แบบเคลื่อนที่ มาตรฐานของระบบเอเจนท์แบบเคลื่อนที่ โปรโตคอลในการเคลื่อนย้ายเอเจนท์ ประเด็นด้านความปลอดภัยของระบบเอเจนท์แบบเคลื่อนที่และระบบเอเจนท์แบบเคลื่อนที่สำหรับการประมูลอิเล็กทรอนิกส์

2.1 คำนิยามที่เกี่ยวข้อง

Object Management Group (OMG) ซึ่งเป็นองค์กรสากลที่ก่อตั้งขึ้นในปี ค.ศ. 1989 ประกอบด้วยสมาชิกที่เป็นทั้งตัวแทนจำหน่ายระบบสารสนเทศ นักพัฒนาซอฟต์แวร์ และผู้ใช้งาน ได้เสนอทฤษฎีและแนวปฏิบัติสำหรับการพัฒนาซอฟต์แวร์ด้วยเทคโนโลยีเชิงวัตถุ ได้ให้คำจำกัดความของคำที่เกี่ยวข้องกับเอเจนท์ไว้ใน Mobile Agent Facility Specification Version 1.0 (OMG 2000) ไว้ดังนี้

2.1.1 เอเจนท์

เอเจนท์ คือ โปรแกรมคอมพิวเตอร์ซึ่งทำงานอัตโนมัติแทนคนหรือองค์กร โดยแต่ละเอเจนท์จะมีเธรด (thread) การประมวลผลของตนเองซึ่งสามารถเริ่มการทำงานเองได้ ทั้งนี้ OMG ได้แบ่งประเภทของเอเจนท์ไว้เป็น 2 ประเภท คือ เอเจนท์แบบสเตชันนารี่และเอเจนท์แบบเคลื่อนที่

2.1.2 เอเจนท์แบบสเตชันนารี่

เอเจนท์แบบสเตชันนารี่ คือ เอเจนท์ที่จะทำการประมวลผลได้เฉพาะบนระบบที่สร้างเอเจนท์นั้น หากเอเจนท์ต้องการติดต่อกับเอเจนท์ที่อยู่ในระบบอื่นก็จะต้องทำการติดต่อผ่านกลไกการติดต่อสื่อสารด้วยการเรียกใช้ส่วนการทำงานระยะไกล (Remote Procedure Call หรือ RPC)

2.1.3 เอเจนท์แบบเคลื่อนที่

เอเจนท์แบบเคลื่อนที่ คือ เอเจนท์ที่ไม่ถูกผูกติดกับระบบที่เอเจนท์เริ่มต้นทำการประมวลผล โดยสามารถเคลื่อนที่ตัวมันเองไปยังระบบอื่นในเครือข่ายได้ ความสามารถในการเคลื่อนที่นี้ทำให้เอเจนท์เคลื่อนย้ายไปยังระบบเอเจนท์ปลายทางที่มีวัตถุที่เอเจนท์ต้องการติดต่อด้วยได้ และเอเจนท์อาจใช้บริการของวัตถุนั้นได้

2.1.4 คุณสมบัติของเอเจนท์แบบเคลื่อนที่

(Wooldridge and Jennings 1995) กล่าวถึงคุณสมบัติพื้นฐานของเอเจนท์ไว้ดังนี้

- **autonomy** เอเจนท์สามารถทำงานได้อัตโนมัติโดยไม่จำเป็นต้องติดต่อโดยตรงกับผู้ใช้ตลอดเวลา และมีกลไกบางอย่างในการควบคุมการกระทำและสถานะของเอเจนท์
- **social ability** เอเจนท์ติดต่อกับเอเจนท์อื่นผ่านภาษาในการติดต่อสื่อสารระหว่างเอเจนท์
- **reactivity** เอเจนท์สามารถทำการตอบสนองเมื่อมีการเปลี่ยนแปลงเกิดขึ้น
- **pro-activeness** เอเจนท์สามารถริเริ่มแสดงพฤติกรรมเพื่อทำงานตามเป้าหมาย

ส่วนคุณสมบัติการเคลื่อนที่ (mobility) นั้นเป็นคุณสมบัติที่เพิ่มขึ้นของเอเจนท์ที่จะทำให้เอเจนท์สามารถเดินทางไปในเครือข่ายได้อัตโนมัติ คุณสมบัตินี้จึงเป็นคุณสมบัติเฉพาะของเอเจนท์แบบเคลื่อนที่ซึ่งความสามารถในการเคลื่อนที่เป็นความต้องการขั้นพื้นฐานของการทำงานด้วยการใช้เอเจนท์ในระบบแบบกระจายโดยเอเจนท์สามารถทำการร้องขอไปยังคอมพิวเตอร์ปลายทางที่ต้องการไปประมวลผล โดยนำสถานะการทำงานรวมถึงรหัสคำสั่งและข้อมูลบางอย่างที่จำเป็นต้องใช้ไปด้วย

โดยสรุปแล้ว เอเจนท์แบบเคลื่อนที่ คือ โปรแกรมคอมพิวเตอร์ซึ่งห่อหุ้มรหัสคำสั่ง ข้อมูล และสถานะการประมวลผลที่สามารถย้ายการประมวลผลไปยังเครื่องคอมพิวเตอร์อื่นที่อยู่ใน

เครือข่ายได้อัตโนมัติ โดยสามารถติดต่อสื่อสารกับเอเจนท์อื่นและตอบสนองต่อสภาวะแวดล้อม เพื่อทำงานตามเป้าหมายของคนหรือองค์กรที่เป็นเจ้าของเอเจนท์นั้น การประมวลผลโดยอาศัยเอเจนท์แบบเคลื่อนที่เป็นลักษณะการประมวลผลแบบกระจาย (distributed computing)

2.2 แนวคิดเรื่องเอเจนท์แบบเคลื่อนที่

เอเจนท์แบบเคลื่อนที่พัฒนามาจากการออกแบบและพัฒนาระบบแบบกระจายโดยมีจุดเริ่มต้นมาจากแนวทางแบบไคลเอนต์-เซิร์ฟเวอร์(client-server paradigm) ซึ่งไคลเอนต์กับเซิร์ฟเวอร์จะทำการติดต่อกันผ่านการส่งข้อความ (message passing) หรือการเรียกใช้ส่วนการทำงานระยะไกลโดยการส่งข้อมูลเป็นพารามิเตอร์ไปให้แก่ส่วนการทำงานเพื่อประมวลผลบนเซิร์ฟเวอร์ ส่วนอีกแนวทางหนึ่งคือ Remote Evaluation (REV) ซึ่งในการทำงานนั้นไคลเอนต์จะทำการส่งรหัสคำสั่งที่จะใช้ในการทำงานไปให้แก่เซิร์ฟเวอร์ เมื่อเซิร์ฟเวอร์ประมวลผลแล้วก็จะส่งผลลัพธ์กลับคืนมาให้แก่ไคลเอนต์ หลังจากนั้นจึงเกิดแนวคิดเรื่องเอเจนท์แบบเคลื่อนที่ซึ่งเป็นโปรแกรมที่ห่อหุ้มรหัสคำสั่งของโปรแกรม ข้อมูล และสถานะการประมวลผลซึ่งสามารถย้ายการประมวลผลไปยังเครื่องคอมพิวเตอร์อื่นที่อยู่ในเครือข่ายได้โดยอัตโนมัติ (Karnik and Tripathi 1998)

เมื่อเปรียบเทียบกับการเรียกส่วนการทำงานระยะไกลซึ่งไคลเอนต์ส่งข้อมูลเป็นพารามิเตอร์ไปประมวลผลที่เซิร์ฟเวอร์จะมีลักษณะการทำงานเป็นแบบซิงโครนัส (synchronous) โดยเมื่อไคลเอนต์ส่งการร้องขอบริการจากเซิร์ฟเวอร์ ไคลเอนต์จะต้องหยุดการทำงานชั่วคราวเพื่อรอการตอบสนองจากเซิร์ฟเวอร์ก่อนจึงจะสามารถทำงานต่อได้ และหากมีการร้องขอบริการซ้ำ ๆ สำหรับงานประยุกต์บางอย่าง ทั้งไคลเอนต์และเซิร์ฟเวอร์ต้องมีการใช้เครือข่ายตลอดเวลา การใช้เอเจนท์แบบเคลื่อนที่จะช่วยลดการใช้แบนด์วิธของเครือข่ายโดยเอเจนท์แบบเคลื่อนที่ไม่จำเป็นต้องใช้เครือข่ายในขณะที่เอเจนท์อยู่ที่เซิร์ฟเวอร์และทำการประมวลผลจนกว่าจะมีการย้ายการทำงานหรือมีความจำเป็นที่จะต้องใช้งานเครือข่ายเท่านั้น ทำให้เอเจนท์แบบเคลื่อนที่เหมาะสมที่จะใช้กับเครือข่ายที่มีแบนด์วิธต่ำ นอกจากนี้ยังเหมาะสมกับระบบที่ต้องมีการใช้งานหรือแลกเปลี่ยนข้อมูลปริมาณมากเพราะสามารถส่งเอเจนท์แบบเคลื่อนที่ไปประมวลผลยังเครื่องคอมพิวเตอร์ที่จัดเก็บข้อมูลแทนการส่งข้อมูลนั้นผ่านเครือข่ายไปประมวลผล

2.3 ภาษาในการพัฒนาระบบเอเจนท์แบบเคลื่อนที่

เนื่องจากระบบเอเจนท์แบบเคลื่อนที่นั้นเป็นระบบที่มีการส่งเอเจนท์ไปประมวลผลยังเซิร์ฟเวอร์อื่น ๆ ในเครือข่ายซึ่งอาจมีระบบปฏิบัติการต่างกัน ดังนั้นจึงต้องการภาษาที่เป็นอิสระจากระบบปฏิบัติการเพื่อให้สามารถประมวลผลได้กับระบบที่แตกต่างกัน และภาษาจะต้องสนับสนุนการเคลื่อนย้ายรหัสคำสั่งในการประมวลผลและมีความปลอดภัยด้วย เช่น มีการสนับสนุนการตรวจสอบชนิดข้อมูล การห่อหุ้มข้อมูลและการเข้าถึงหน่วยความจำโดยมีการระบุขนาดหน่วยความจำ เป็นต้น

ในการพัฒนาเอเจนท์แบบเคลื่อนที่นั้นสามารถใช้ภาษาในการเขียนโปรแกรมได้หลายภาษา ตัวอย่างของระบบเอเจนท์แบบเคลื่อนที่ที่ได้มีการนำเสนอในงานวิจัย เช่น ระบบ Telescript ซึ่งพัฒนาด้วยภาษา Telescript ระบบ Agent Tcl (Fuggetta et. al 1998) พัฒนาด้วยภาษา Tcl ซึ่งเป็นภาษาสคริปต์ แม้ว่าจะสะดวกในการสร้างโปรแกรมเอเจนท์ขนาดกลาง แต่ก็มีข้อเสียคือการแบ่งส่วนการทำงานเป็นโมดูล และการห่อหุ้มข้อมูลยังไม่มีประสิทธิภาพ ส่วนการใช้ภาษาจาวามีข้อดีคือ สามารถสร้างระบบเอเจนท์ที่ซับซ้อนได้ง่ายและเอเจนท์ที่พัฒนาสามารถนำมาใช้งานบนระบบปฏิบัติการที่แตกต่างกันโดยไม่ต้องทำการแก้ไขหรือทำการแปลโปรแกรมใหม่ ตัวอย่างระบบที่พัฒนาด้วยภาษาจาวา เช่น Aglets (Lange and Oshima 2003), JADE (Tilab 2004), FIPA-OS (Poslad et. al 2000) เป็นต้น

คุณสมบัติที่ทำให้ภาษาจาวาเป็นภาษาที่ดีสำหรับการเขียนโปรแกรมเอเจนท์แบบเคลื่อนที่ (Lange and Oshima 2003) มีดังนี้

1. โปรแกรมประยุกต์ที่เขียนด้วยภาษาจาวานั้นเป็นอิสระจากแพลตฟอร์ม โดยสามารถนำไปประมวลผลบนระบบปฏิบัติการได้ทุกระบบ
2. ทุกครั้งที่ทำการแปลโปรแกรม ตัวแปลภาษาจาวาจะทำหน้าที่ตรวจสอบรหัสคำสั่งต้นฉบับให้ตรงกับกฎความปลอดภัยของภาษา เช่น การห้ามใช้คำสั่ง goto และพอยน์เตอร์ (pointer) การห้ามติดต่อกับหน่วยความจำโดยตรง และเมื่อมีการเรียกใช้โปรแกรมจาวาในรูปแบบรหัสไบต์ (bytecode) ลงมาที่เครื่องคอมพิวเตอร์ โปรแกรมเหล่านั้นจะต้องผ่านการตรวจสอบจากตัวตรวจสอบรหัสไบต์ (Bytecode Verifier) ว่ารหัสไบต์ดังกล่าวไม่ได้ถูกแก้ไขเพิ่มเติมหลังจากที่มีการแปล และมีโครงสร้างถูกต้องตรงตามไวยากรณ์และกฎความปลอดภัยของภาษาจาวา

3. การเข้าถึงทรัพยากรที่สำคัญของระบบนั้นจะทำผ่าน JVM (Java Virtual Machine) และจะมีการตรวจสอบเพื่อควบคุมการทำงานของโปรแกรมภาษาจาวา เช่น ไม่ให้ใช้คำสั่งของระบบปฏิบัติการ และไม่ให้มีการเปิดไดรเวอร์ของอุปกรณ์ระบบของฮาร์ดดิสก์ เป็นต้น
4. ภาษาจาวามีกลไกในการป้องกันการเขียนทับบนหน่วยความจำและการแก้ไขข้อมูล มีการตรวจสอบชนิดข้อมูลในการคำนวณทางคณิตศาสตร์และไม่อนุญาตให้ใช้พอยน์เตอร์ โปรแกรมไม่สามารถปลอมตัวเพื่อเข้าถึงข้อมูลที่ไม่มีสิทธิ์เพื่อเป็นการป้องกันไวรัส ดังนั้นหากมีการแก้ไขรหัสไบต์สถาปัตยกรรมความปลอดภัยของจาวาจะป้องกันไม่ให้นำรหัสไบต์ไปประมวลผลได้
5. มีกลไกการดึงคลาสมาใช้งานแบบไดนามิค (dynamic class loading) ซึ่งเป็นกลไกที่อนุญาตให้ JVM ทำการดึงคลาสในขณะประมวลผลซึ่งป้องกันให้แต่ละเอเจนท์ให้ประมวลผลเป็นอิสระกันและปลอดภัยจากเอเจนท์อื่น
6. รองรับการเขียนโปรแกรมแบบมัลติเธรด (multithread programming) เนื่องจากเอเจนท์นั้นต้องสามารถประมวลผลเป็นอิสระจากเอเจนท์อื่นที่อยู่ในที่เดียวกัน การอนุญาตให้เอเจนท์ประมวลผลด้วยเธรดของตนเองเป็นวิธีการที่ทำให้เอเจนท์สามารถทำงานได้อย่างอิสระ
7. คุณสมบัติหลักของเอเจนท์แบบเคลื่อนที่ก็คือสามารถเคลื่อนย้ายไปพร้อมกับสถานะ รหัสคำสั่ง และข้อมูลซึ่งถูกแปลงให้อยู่ในรูปแบบที่สามารถส่งไปในเครือข่ายแล้วสามารถทำงานต่อเมื่อเอเจนท์เดินทางมาถึงปลายทางได้ ซึ่งภาษาจาวามีกลไกการ serialization อยู่ในภาษาแล้วเพื่อรองรับการทำงานในส่วนนี้ให้แก่ระบบเอเจนท์

2.4 มาตรฐานของระบบเอเจนท์แบบเคลื่อนที่

เนื่องจากอุตสาหกรรมการผลิตซอฟต์แวร์มีการเจริญเติบโตอย่างรวดเร็ว ทำให้มีการพัฒนาระบบเอเจนท์แบบเคลื่อนที่ออกมาหลากหลายระบบ ดังนั้นจึงจำเป็นต้องมีการกำหนดคุณลักษณะบางอย่างที่จำเป็นสำหรับระบบเอเจนท์แบบเคลื่อนที่ให้เป็นมาตรฐานเพื่อทำให้ระบบเอเจนท์แบบเคลื่อนที่ของผู้ผลิตที่แตกต่างกันสามารถทำงานร่วมกันได้ มาตรฐานของระบบเอเจนท์แบบเคลื่อนที่ได้แก่ มาตรฐาน OMG MASIF (OMG 2000) และ FIPA (FIPA 2002)

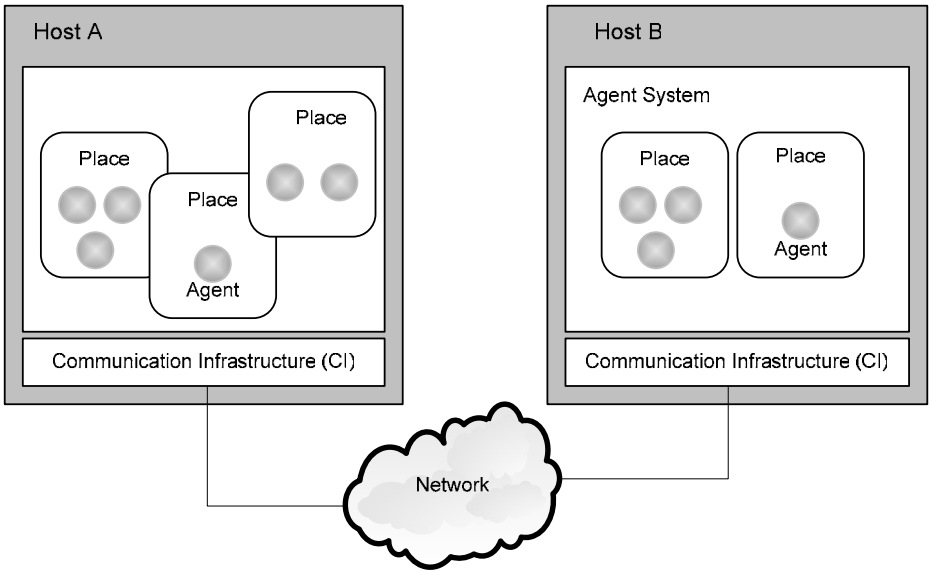
2.4.1 มาตรฐาน OMG MASIF

OMG Mobile Agent System Interoperability Facility (MASIF) เกิดขึ้นจากบริษัทชั้นนำ 5 บริษัทคือ Crystaliz Inc., General Magic Inc., GMD FOKUS, IBM และ Open Group ที่ต้องการให้ระบบเอเจนท์ต่างผู้ผลิตกันสามารถทำงานร่วมกันได้ โดยมาตรฐานนี้กล่าวถึงการติดต่อสื่อสารระหว่างระบบเอเจนท์แบบเคลื่อนที่และการติดต่อสื่อสารระหว่างระบบเอเจนท์แบบเคลื่อนที่กับระบบอื่นโดยใช้ Common Object Request Broker Architecture (CORBA) แต่ MASIF ไม่ได้กล่าวถึงภาษาที่จะใช้ร่วมกันได้ดังนั้น MASIF จึงเป็นข้อกำหนดสำหรับเฉพาะระบบเอเจนท์ที่ใช้ภาษาเดียวกันแต่ต่างผู้ผลิตกัน

มาตรฐาน OMG MASIF ประกอบด้วยข้อกำหนดในเรื่องต่อไปนี้

- **การจัดการเอเจนท์ (Agent Management)** เจ้าของระบบเอเจนท์ต้องสามารถจัดการกับเอเจนท์ได้ เช่น การค้นหาเอเจนท์ การหยุดการทำงานของเอเจนท์ สั่งให้เอเจนท์ทำงานได้ เป็นต้น
- **การเคลื่อนย้ายเอเจนท์ (Agent Transfer)** โปรแกรมประยุกต์ต้องสามารถส่งเอเจนท์ไปยังระบบเอเจนท์อื่น ๆ ได้
- **ชื่อเอเจนท์และชื่อระบบเอเจนท์ (Agent and Agent System Names)** ไวยากรณ์และความหมายที่ใช้ในการอ้างอิงจะต้องเป็นมาตรฐานเพื่อให้ระบบเอเจนท์สามารถติดต่อสื่อสารกันได้
- **ไวยากรณ์ของตำแหน่งและชนิดของระบบ (Agent System type and location syntax)** ไวยากรณ์ของตำแหน่งของระบบเอเจนท์ต้องเป็นมาตรฐานเพื่อให้เอเจนท์สามารถเข้าถึงข้อมูลของระบบเอเจนท์ปลายทางที่เอเจนท์จะทำการเคลื่อนย้ายไปทำงานได้ และชื่อของระบบเอเจนท์ต้องเป็นไปตามมาตรฐานและไม่ซ้ำกัน

แบบจำลองระบบเอเจนท์ตามมาตรฐาน MASIF แสดงดังภาพประกอบ 2.1



ภาพประกอบ 2.1 แบบจำลองระบบเอเจนท์ตามมาตรฐาน MASIF (OMG 2000)

แบบจำลองระบบเอเจนท์ตามมาตรฐาน MASIF ประกอบด้วย

- **ระบบเอเจนท์ (agent system)** เป็นแพลตฟอร์มสำหรับเอเจนท์ที่ทำงานอยู่บนเครื่องคอมพิวเตอร์เพื่อสร้าง ประมวลผล เคลื่อนย้าย และทำลายเอเจนท์ โดยระบบเอเจนท์จะมีชื่อและตำแหน่งบนเครือข่ายไม่ซ้ำกัน และในเครื่องคอมพิวเตอร์หนึ่งเครื่องอาจมีหลายระบบเอเจนท์
- **Place** เป็นสภาวะแวดล้อมที่เอเจนท์สามารถทำการประมวลผลได้ภายในระบบเอเจนท์
- **Communication Infrastructure (CI)** เป็นโครงสร้างพื้นฐานในการติดต่อระหว่างระบบเอเจนท์ซึ่งรองรับการติดต่อสื่อสารโดยการใช้ RMI, CORBA และ Distributed Component Object Model (DCOM) รวมทั้งยังมีบริการสำหรับการเคลื่อนที่ของเอเจนท์ด้วย

เมื่อมีการนำหลายระบบเอเจนท์มาใช้งานในองค์กรเดียวกันจะทำให้เกิดเป็นเครือข่ายของระบบเอเจนท์ขององค์กรเรียกว่า Region ในระบบเอเจนท์ของ MASIF นั้นมีอินเตอร์เฟสหลัก 2 อินเตอร์เฟสคือ MAFAgentSystem และ MAFFinder เพื่อให้ระบบเอเจนท์ที่ต่างกันสามารถทำงานร่วมกันได้ โดย MAFAgentSystem เป็น Application Programming Interface (API) สำหรับการจัดการเอเจนท์และการจัดการ Place ส่วน MAFFinder ทำหน้าที่ให้บริการชื่อสำหรับเอเจนท์ Place และระบบเอเจนท์ ซึ่งได้แก่การลงทะเบียนและการระบุตำแหน่งของเอเจนท์

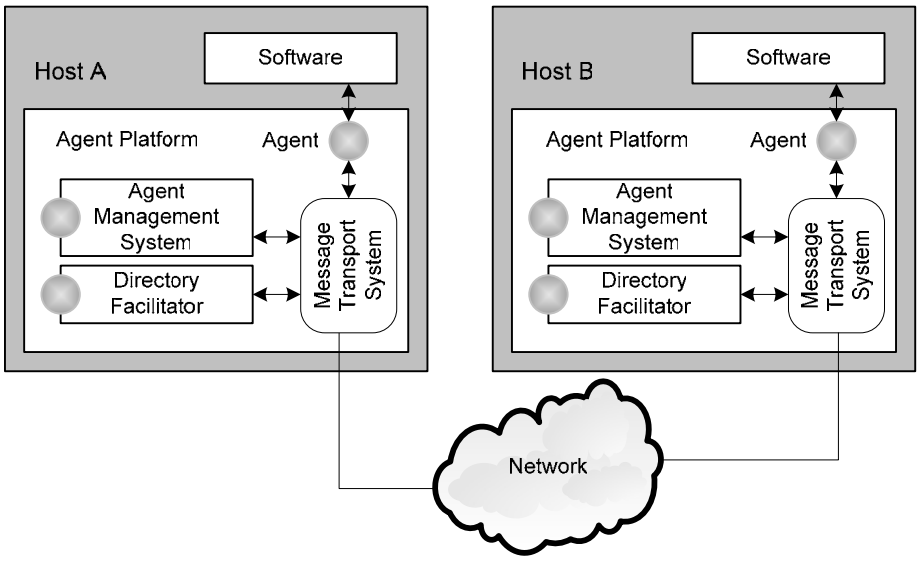
Place และระบบเอเจนท์ นอกจากนี้การติดต่อเข้ามาใน Region จะต้องทำผ่าน Region Access Point (RAP) ซึ่งเพื่อตรวจสอบเอเจนท์ที่เข้าสู่ระบบว่าเป็นเอเจนท์ที่ได้รับสิทธิ์ในการใช้งานใน Region

2.4.2 มาตรฐาน FIPA

The Foundation For Intelligent Physical Agents (FIPA) ก่อตั้งขึ้นในปีค.ศ. 1996 เพื่อสร้างมาตรฐานซอฟต์แวร์สำหรับระบบเอเจนท์ และในปีค.ศ. 2005 องค์กรมาตรฐาน IEEE Computer Society ได้มีมติรับ FIPA เป็นหนึ่งในคณะกรรมการมาตรฐานของ IEEE โดย FIPA ได้เสนอข้อกำหนด Foundation For Intelligent Physical Agents Specifications (FIPA 2002) ซึ่งเป็นข้อกำหนดที่เกี่ยวข้องกับการติดต่อสื่อสารระหว่างเอเจนท์ การส่งข้อความระหว่างเอเจนท์ การจัดการเอเจนท์ แนวคิดสำหรับสถาปัตยกรรมและโปรแกรมประยุกต์สำหรับระบบเอเจนท์

2.4.2.1 แบบจำลองการจัดการเอเจนท์

องค์ประกอบระบบเอเจนท์ที่เสนอในข้อกำหนดตามมาตรฐานของ FIPA แสดงด้วยแบบจำลองการจัดการเอเจนท์(Agent Management Reference Model) ดังภาพประกอบ 2.2 ซึ่งแบบจำลองนี้ได้ถูกนำไปพัฒนาเป็นเอเจนท์แพลตฟอร์ม (Agent Platform :AP) หรือระบบเอเจนท์นั่นเอง



ภาพประกอบ 2.2 แบบจำลองการจัดการเอเจนท์ตามมาตรฐาน FIPA (FIPA 2002)

องค์ประกอบของระบบเอเจนท์ในแบบจำลองการจัดการเอเจนท์ตามมาตรฐาน FIPA ประกอบด้วย

- **Agent Platform (AP)**

AP เป็นระบบเอเจนท์ที่มีโครงสร้างพื้นฐานทางกายภาพที่ เอเจนท์สามารถทำงานได้ โดยระบบเอเจนท์ประกอบด้วยเครื่องคอมพิวเตอร์ ระบบปฏิบัติการ ซอฟต์แวร์สนับสนุนเอเจนท์ องค์ประกอบในการจัดการเอเจนท์และเอเจนท์ การออกแบบภายในระบบเอเจนท์เป็นประเด็นสำหรับผู้พัฒนาระบบเอเจนท์และไม่ได้กำหนดไว้เป็นมาตรฐานใน FIPA โดยเอเจนท์ที่อยู่ในระบบเอเจนท์เดียวกันไม่จำเป็นต้องอยู่บนคอมพิวเตอร์เครื่องเดียวกันก็ได้ ทั้งนี้แต่ละระบบเอเจนท์จะทำงานโดยใช้หนึ่งโพรเซส และแต่ละเอเจนท์จะทำงานโดยใช้เธรด

- **Agent Management System (AMS)**

AMS เป็นเอเจนท์ที่ทำการควบคุมการเข้าถึงและการใช้งานระบบเอเจนท์โดยภายในหนึ่งระบบเอเจนท์จะมีได้เพียงหนึ่ง AMS เท่านั้น หน้าที่ของ AMS ได้แก่ การเก็บรายการ Agent Identifiers (AIDs) ของเอเจนท์ที่ลงทะเบียนในระบบ การจัดการการดำเนินการเกี่ยวกับเอเจนท์ เช่น การสร้างเอเจนท์ การลบเอเจนท์และการย้ายเอเจนท์เข้ามาหรือออกจากระบบเอเจนท์อื่น การสอบถามรายละเอียดของระบบเอเจนท์อื่นก่อนที่จะดำเนินการย้ายเอเจนท์ และการดูแลวงจรชีวิตของเอเจนท์ที่ลงทะเบียนไว้กับระบบเอเจนท์

- **Directory Facilitator (DF)**

DF เป็นเอเจนท์ที่ให้บริการไดเร็คทอรี (directory) สำหรับการสืบค้นบริการของเอเจนท์ ทำหน้าที่ในการเก็บรักษารายการบริการของเอเจนท์และให้บริการข้อมูลเกี่ยวกับเอเจนท์ที่อยู่ในไดเร็คทอรี แก่เอเจนท์อย่างเท่าเทียมกันโดยเอเจนท์สามารถลงทะเบียนบริการของตนไว้กับ DF หรือสอบถาม DF ว่าเอเจนท์อื่นมีบริการอะไรบ้าง ซึ่งแต่ละระบบเอเจนท์อาจมีหลาย DF ร่วมกันทำงาน เอเจนท์ที่ต้องการให้บริการแก่เอเจนท์อื่นต้องลงทะเบียนรายละเอียดบริการของเอเจนท์ไว้กับ DF แต่เอเจนท์อาจจะปฏิเสธการให้บริการได้หลังจากที่ทำการลงทะเบียนไว้แล้ว ดังนั้น DF จึงไม่สามารถรับรองความเป็นปัจจุบันของข้อมูลที่ลงทะเบียนไว้ นอกจากนี้เอเจนท์อาจทำการถอนการลงทะเบียนที่ DF หรือแก้ไขรายละเอียดเอเจนท์ได้ในภายหลัง

- **Message Transport System (MTS)**

MTS ทำหน้าที่ในส่งข้อความระหว่างเอเจนท์โดยการรับส่งข้อความระหว่างเอเจนท์จะทำผ่าน MTS เท่านั้น ดังนั้น

- **เอเจนท์ (Agent)**

เอเจนท์เป็นโปรแกรมคอมพิวเตอร์ที่สามารถทำงานแบบอัตโนมัติและติดต่อสื่อสารกับโปรแกรมประยุกต์เพื่อทำงานตามที่กำหนด เอเจนท์ติดต่อสื่อสารกันโดยใช้ภาษาในการติดต่อสื่อสารของเอเจนท์(Agent Communication Language หรือ ACL) และอาจมีบริการสำหรับให้เอเจนท์อื่นสามารถเรียกใช้งานได้ นอกจากนี้เอเจนท์ต้องมีเจ้าของเอเจนท์(owner) ซึ่งอาจเป็นคนหรือองค์กร และมีชื่อเอเจนท์เพื่อใช้ในการอ้างถึง (Agent Identifier หรือ AID) และติดต่อกับเอเจนท์อื่น

2.4.2.2 การกำหนดชื่อเอเจนท์ (Agent Naming)

เอเจนท์ต้องมีการกำหนดชื่อที่ไม่ซ้ำกันเพื่อให้สามารถถูกระบุหรือเรียกใช้งานได้เมื่อต้องการ เช่น เจ้าของเอเจนท์สามารถติดต่อหรือควบคุมเอเจนท์ได้เมื่อมีการส่งเอเจนท์ไปประมวลผลในเครือข่าย รูปแบบของชื่อเอเจนท์ในแบบจำลองการอ้างอิงของ FIPA นั้นมีรูปแบบเป็นคู่ของพารามิเตอร์-ค่า (parameter-value pairs) ที่เรียกว่า Agent Identifier (AID) ซึ่ง AID นี้จะไม่สามารถแก้ไขได้ตลอดระยะเวลาที่เอเจนท์ยังมีชีวิตอยู่ พารามิเตอร์สำหรับการกำหนดชื่อเอเจนท์แสดงดังตาราง 2.1

ตาราง 2.1 พารามิเตอร์สำหรับการกำหนดชื่อเอเจนท์ (FIPA 2002)

พารามิเตอร์	คำอธิบาย
name	ค่าบ่งบอกที่ไม่ซ้ำ (unique identifier) ซึ่งสามารถอ้างอิงถึงได้ กลไกที่ง่ายที่สุดคือสร้างจากชื่อจริง (actual name) ของเอเจนท์และ home agent platform (HAP) address หรือที่อยู่ของเอเจนท์แพลตฟอร์มที่สร้างเอเจนท์นั้นโดยแยกด้วยอักขระ @ เช่น *name agent-b@bar.com*
addresses	รายการของที่อยู่ทางกายภาพที่เอเจนท์สามารถใช้ในการติดต่อสื่อสารกัน (transport addresses) เช่น *addresses (sequence iiop://bar.com/acc)*
resolvers	รายการของ name resolution service addresses เช่น *(sequence (agent-identifier* *:name ams@foo.com* *:addresses (sequence iiop://foo.com/acc)))*

ตัวอย่าง AID ที่สอดคล้องกับตาราง 2.1 เช่น

(agent-identifier

:name agent-b@bar.com

addresses (sequence iiop://bar.com/acc)

:resolvers (sequence

(agent-identifier

:name ams@foo.com

:addresses (sequence iiop://foo.com/acc))))

2.4.2.3 Name Resolution

Name resolution เป็นกลไกที่ระบบใช้เพื่อหาตำแหน่งปัจจุบันของวัตถุ (Karnik and Tripathi 1998) โดยในระบบเอเจนท์แบบเคลื่อนที่นั้น name resolution เป็นบริการของ AMS ซึ่งเอเจนท์อื่นสามารถร้องขอให้ AMS ค้นหา ตัวอย่างรูปแบบของ name resolution เป็นดังนี้

(1) เอเจนท์ a ต้องการส่งข้อความให้เอเจนท์ b ซึ่งมี AID เป็นดังนี้

(agent-identifier

:name agent-b@bar.com

:resolvers (sequence

(agent-identifier

:name ams@foo.com

:addresses (sequence iiop://foo.com/acc))))

และเอเจนท์ a ต้องการรู้ที่อยู่ของเอเจนท์ b

(2) ดังนั้นเอเจนท์ a สามารถส่งการร้องขอการค้นหาไปยังเอเจนท์แรกในพารามิเตอร์ resolvers ซึ่งปกติคือ AMS ซึ่งในตัวอย่างนี้คือ AMS ที่อยู่ใน foo.com

(3) ถ้า AMS ที่ foo.com มีเอเจนท์ b ลงทะเบียนอยู่ก็จะส่งข้อความที่มีที่อยู่เอเจนท์ b กลับไปให้เอเจนท์ a แต่ถ้าไม่มีเอเจนท์ b ลงทะเบียนอยู่ก็จะส่งข้อความกลับไปให้เอเจนท์ a ว่าไม่มี

(4) เมื่อเอเจนท์ a ได้รับข้อความกลับมา ก็จะแยกข้อความออกมาซึ่งจะทำให้ทราบที่อยู่ที่อยู่ของเอเจนท์ b

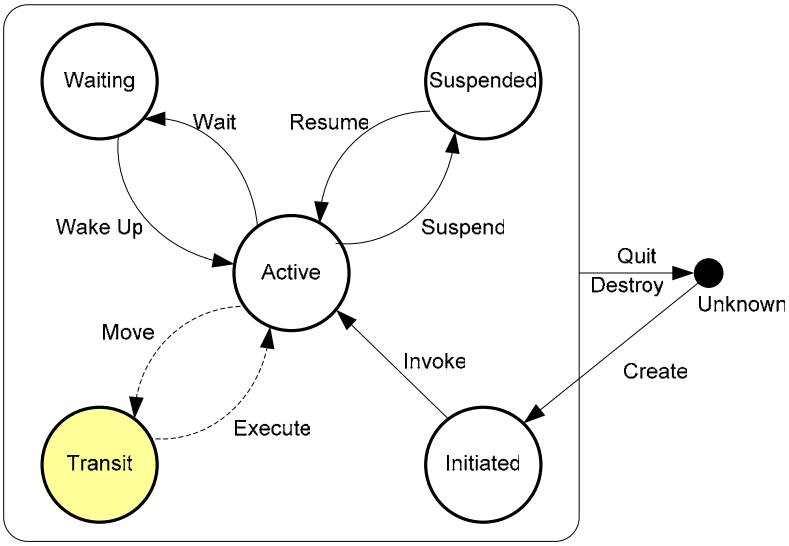
(5) เอเจนท์ a สามารถส่งข้อความให้แก่เอเจนท์ b ได้โดยการเพิ่มพารามิเตอร์ address ใน AID ของเอเจนท์ b

2.4.2.4 วงจรชีวิตของเอเจนท์ (Agent Life Cycle)

เมื่อเอเจนท์ถูกสร้างขึ้นมาอยู่ในระบบเอเจนท์และใช้บริการต่าง ๆ ของระบบเอเจนท์ ทั้งนี้เอเจนท์ซึ่งเป็นโปรเซสของซอฟต์แวร์จะมีวงจรชีวิตที่ถูกจัดการโดยระบบเอเจนท์ตามกฎต่อไปนี้

- **AP Bounded** เอเจนท์จะถูกผูกติดกับระบบเอเจนท์ โดยเอเจนท์จะถูกจัดการทางกายภาพภายในระบบเอเจนท์หนึ่งเสมอ และวงจรชีวิตของเอเจนท์จะผูกติดกับระบบเอเจนท์นั้น
- **Application Independent** แบบจำลองวงจรชีวิตของเอเจนท์ (agent life cycle model) เป็นอิสระจากระบบโปรแกรมประยุกต์และถูกมองเห็นเฉพาะสถานะและการเปลี่ยนสถานะของเอเจนท์ในวงจรชีวิตของเอเจนท์เท่านั้น
- **Instance-Oriented** เอเจนท์ที่ถูกอธิบายในแบบจำลองวงจรชีวิตของเอเจนท์จะถูกมองว่าเป็นวัตถุที่ถูกสร้างขึ้นซึ่งจะมีชื่อที่ไม่ซ้ำกันและทำการประมวลผลอย่างอิสระ
- **Unique** แต่ละเอเจนท์มีสถานะของวงจรชีวิตได้เพียงหนึ่งสถานะ ณ ขณะเวลาหนึ่งและภายในหนึ่ง AP

วงจรชีวิตของเอเจนท์ตามมาตรฐาน FIPA แสดงได้ดังภาพประกอบ 2.3



ภาพระกอบ 2.3 วงจรชีวิตของเอเจนท์ตามมาตรฐาน FIPA (FIPA 2002)

จากภาพประกอบ 2.4 วงจรชีวิตของเอเจนท์จะเริ่มขึ้นเมื่อเอเจนท์ถูกสร้าง (Create) ขึ้นโดยระบบเอเจนท์ เริ่มต้นเอเจนท์จะมีสถานะเป็น Initiated จนกว่าระบบเอเจนท์จะทำการ Invoke จึงจะเข้าสู่สถานะ Active ซึ่งเอเจนท์อาจจะมีการดำเนินการเพื่อเปลี่ยนสถานะในลักษณะ Suspend, Wait และ Move เพื่อเข้าสู่สถานะ Suspended, Waiting และ Transit ตามลำดับ หลังจากนั้นเอเจนท์สามารถกลับสู่สถานะ Active ได้อีกครั้งด้วยการ Resume, Wake Up และ Execute ตามลำดับ ทั้งนี้การทำงานของเอเจนท์จะสิ้นสุดลงเมื่อระบบเอเจนท์สั่ง Quit หรือ Destroy โดยสถานะ Transit และการดำเนินการเปลี่ยนสถานะ Move และ Execute ซึ่งแสดงด้วยเส้นประจะเกิดกับเอเจนท์แบบเคลื่อนที่เท่านั้น สรุปการเปลี่ยนสถานะ (transition) ของเอเจนท์ แสดงดังตาราง 2.2

ตาราง 2.2 สรุปการเปลี่ยนสถานะของเอเจนท์

การดำเนินการเพื่อเปลี่ยนสถานะ	**คำอธิบาย**
Create	การสร้างหรือติดตั้งเอเจนท์ใหม่
Invoke	การปลุกให้เอเจนท์ทำงาน
Destroy	การบังคับให้เอเจนท์หยุดการทำงานซึ่งเฉพาะ AMS เท่านั้นที่ทำได้ และเอเจนท์ไม่สามารถปฏิเสธได้
Quit	การขอให้เอเจนท์หยุดการทำงานแต่เอเจนท์อาจไม่ทำก็ได้
Suspend	การทำให้เอเจนท์เข้าสู่สถานะ Suspended โดยเอเจนท์หรือ AMS

ตาราง 2.2 (ต่อ)

การดำเนินการเพื่อเปลี่ยนสถานะ	คำอธิบาย
Resume	ทำให้เอเจนท์ออกจากสถานะ suspended ทำได้โดย AMS เท่านั้น
Wait	การทำให้เอเจนท์เข้าสู่สถานะรอ ทำได้โดยเอเจนท์
Wake Up	ทำให้เอเจนท์ออกจากสถานะรอ ทำได้โดย AMS เท่านั้น
Move*	ทำให้เอเจนท์เข้าสู่สถานะ transit ทำได้โดยเอเจนท์
Execute*	ทำให้เอเจนท์ออกจากสถานะ transit ทำได้โดย AMS

* เป็นการเปลี่ยนสถานะที่ใช้ได้เฉพาะเอเจนท์แบบเคลื่อนที่เท่านั้น

เมื่อเอเจนท์มีการเปลี่ยนสถานะ การส่งข้อความให้แก่เอเจนท์ผ่านทาง MTS ในแต่ละสถานะของวงจรชีวิตของเอเจนท์จะถูกจัดการโดย AMS ซึ่งการส่งข้อความในแต่ละสถานะของเอเจนท์แสดงดังตาราง 2.3

ตาราง 2.3 การส่งข้อความในแต่ละสถานะของเอเจนท์

สถานะ	การส่งข้อความ
Active	MTS ส่งข้อความให้แก่เอเจนท์แบบปกติ
Initiated/Waiting/Suspended	MTS ทำการเก็บข้อความไว้จนกว่าเอเจนท์จะเปลี่ยนสถานะเป็น Active หรือทำการส่งต่อข้อความไปยังตำแหน่งใหม่หากว่ามีการระบุว่าให้ส่งต่อให้เอเจนท์
Transit	MTS ทำการเก็บข้อความไว้จนกว่าเอเจนท์จะเปลี่ยนสถานะเป็น Active ซึ่งอาจเกิดจากการย้ายการทำงานของเอเจนท์ล้มเหลวที่ระบบเอเจนท์ต้นทางหรือเอเจนท์เริ่มย้ายการทำงานไปที่ระบบเอเจนท์ปลายทางแล้ว หรือทำการส่งต่อข้อความไปยังตำแหน่งใหม่หากว่ามีการระบุว่าให้ส่งต่อให้เอเจนท์ สถานะ Transit นี้จะเกิดขึ้นกับเอเจนท์แบบเคลื่อนที่เท่านั้น
Unknown	MTS ทำการเก็บข้อความไว้หรือไม่รับข้อความนั้นขึ้นอยู่กับนโยบายของ MTS และข้อกำหนดในข้อความนั้น

2.4.2.5 การติดต่อสื่อสารระหว่างเอเจนท์

เอเจนท์จะมีการติดต่อสื่อสารกันด้วยการส่งข้อความเพื่อทำงานร่วมกันหรือทำการเจรจาต่อรองกัน เนื่องจากเอเจนท์อาจสร้างมาจากต่างระบบกันจึงต้องมีการกำหนดมาตรฐานโครงสร้างข้อความระหว่างเอเจนท์ซึ่งเกี่ยวข้องกับโดเมนที่เอเจนท์นั้นทำงานเพื่อทำให้ทุกเอเจนท์เข้าใจตรงกัน และแม้ว่าจะไม่เข้าใจเนื้อหาแต่ก็ควรเข้าใจโครงสร้างของข้อความที่เป็นมาตรฐานโครงสร้างนี้เรียกว่า Agent Communication Languages (ACLs) ดังแสดงในตาราง 2.4

ตาราง 2.4 โครงสร้างข้อความ Agent Communication Languages

โครงสร้าง	คำอธิบาย
performative	เป็นคำ 1 คำที่จะอธิบายจุดประสงค์ของข้อความ เช่น แจ้งเพื่อทราบ (tell), ยกเลิก (cancel), ลงทะเบียน (register), ตอบกลับ (reply)
sender	ชื่อและที่อยู่ของเอเจนท์ที่เป็นผู้ส่ง
receiver	ชื่อและที่อยู่ของเอเจนท์ที่เป็นผู้รับ
language	ภาษาที่ใช้ในเนื้อหาของข้อความ
ontology	ออนโทโลยีหรือหรือคำศัพท์เฉพาะในการแปลข้อความ
content	เนื้อหาข้อความที่ต้องการส่ง

2.5 โปรโตคอลในการย้ายเอเจนท์ (Agent Transfer Protocol:ATP)

ATP (Lange and Aridor 1997) เป็นโปรโตคอลในระดับงานประยุกต์เพื่อใช้ในการส่งเอเจนท์บนเครือข่ายซึ่งเป็นอิสระจากภาษาและระบบปฏิบัติการ เครื่องคอมพิวเตอร์ที่ติดตั้งระบบเอเจนท์จะมีบริการที่เรียกว่า เอเจนท์เซอร์วิส (agent service) ทำหน้าที่รับส่งเอเจนท์จากระบบเอเจนท์ที่อยู่บนเครื่องอื่นในเครือข่าย สิ่งที่กำหนดใน ATP ประกอบด้วย

1. Agent Identifier (ID) เป็นส่วนของชุดของตัวเลขเพื่อใช้ในการอ้างถึงเอเจนท์
2. Naming of agent service เป็นส่วนของการตั้งชื่อเอเจนท์เซอร์วิส
3. Agent transportation เป็นส่วนของการส่งเอเจนท์

2.5.1 Naming of Agent Services

ชื่อของเอเจนท์เซอร์วิสระบุด้วย Uniform Resource Identifier (URI) (Berners-Lee et. al 2005) ซึ่งประกอบด้วย Uniform Resource Locators (URL) และ Uniform Resource Name (URN) รูปแบบข้อความของ URI จะทำให้ทราบถึงทรัพยากรของเอเจนท์เมื่อใช้โปรโตคอล ATP

โปรโตคอล ATP จะใช้ ATP URL ในการบ่งบอกถึงทรัพยากรของเอเจนท์ที่อยู่ในเครือข่าย โดย ATP URL มีรูปแบบดังนี้

ATP_URL = *Service_host* [*Agent_resource* | *Class_resource*]

Service_host = "atp://" *Host* [: *Port*]

Host = <An Internet host domain name or IP address>

Port = *Digit+*

Agent_resource = [*Name*] "#" *Agent_identifier*

Name= "/" <a string>

Class_resource = Class_path

Class_path = <A legal absolute path specification>

ตัวอย่างการระบุ ATP URL ด้วยทรัพยากรของเอเจนท์ แสดงดังตาราง 2.5

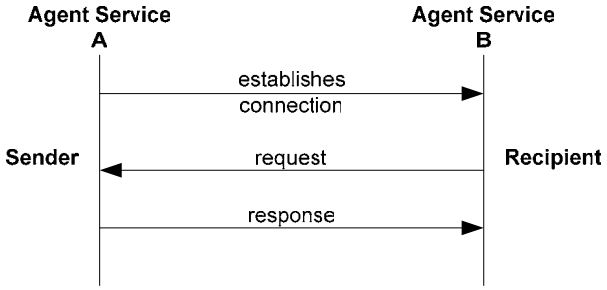
ตาราง 2.5 ตัวอย่างการระบุ ATP URL ด้วยทรัพยากรของเอเจนท์

ATP URL	ความหมาย
atp://joe.ibm.com/28537922	เอเจนท์ที่มี ID= 28537922 บนเอเจนท์เซอร์วิส atp://joe.ibm.com
atp://joe.ibm.com/danny#28537922	เอเจนท์ของ danny ที่มี ID=28537922 บนเอเจนท์เซอร์วิส atp://joe.ibm.com
atp://joe.ibm.com/agents/classes/Hello.class	เอเจนท์ที่มีคลาสคือ Hello.class อยู่ที่โฟลเดอร์ /agents/classes บนเอเจนท์เซอร์วิส atp://joe.ibm.com

2.5.2 Agent Transportation

โปรโตคอล ATP ใช้การส่งข้อความร้องขอและตอบกลับระหว่างเอเจนท์เซอร์วิส โดยเอเจนท์เซอร์วิสที่เป็นฝ่ายส่งการร้องขอทำหน้าที่เป็นผู้ส่ง (sender) ส่วนเอเจนท์เซอร์วิสที่รับ

การร้องขอและตอบกลับทำหน้าที่เป็นผู้รับ (recipient) การส่งข้อความระหว่างเอเจนท์เซอร์วิส แสดงดังภาพประกอบ 2.4



ภาพประกอบ 2.4 การส่งข้อความระหว่างเอเจนท์เซอร์วิส

รายละเอียดของข้อความที่เป็นการร้องขอ (request) และข้อความที่เป็นการตอบรับ (response) ประกอบด้วยพารามิเตอร์ต่าง ๆ แสดงดังตาราง 2.6

ตาราง 2.6 พารามิเตอร์ของข้อมูลการร้องขอและการตอบรับ

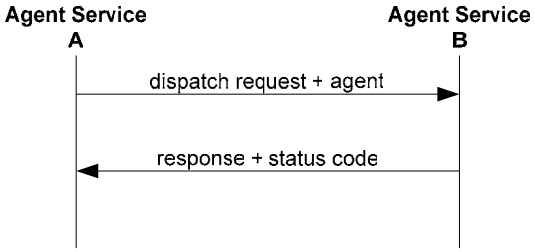
Type	Parameter Group	Parameter
การร้องขอ	Request line	- Request method ได้แก่ dispatch, retract, fetch, message - Protocol version - Required resources
	MIME-like message	- Request modifiers - Sender information - content อาจมีหรือไม่มีก็ได้
การร้องขอ	Status line	- Success/Error code - Protocol version
	MIME-like message	- Response modifier - Sender information - content อาจมีหรือไม่มีก็ได้

2.5.3 รูปแบบการส่งข้อความร้องขอและตอบกลับใน ATP

รูปแบบการส่งข้อความร้องขอและตอบกลับใน ATP มี 4 รูปแบบ คือ

1. Dispatch an agent from A to B

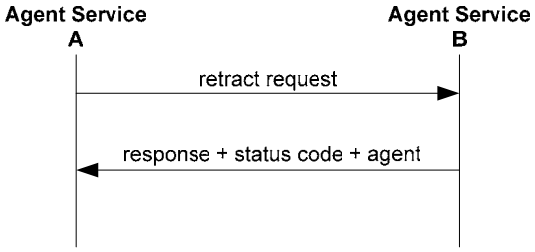
การส่งเอเจนท์จาก A ไป B โดยเอเจนท์จะถูกส่งไปภายในข้อความ dispatch request การส่งข้อความ dispatch request แสดงดังภาพประกอบ 2.5



ภาพประกอบ 2.5 การส่งข้อความ dispatch request

2. Retract an agent from B to A

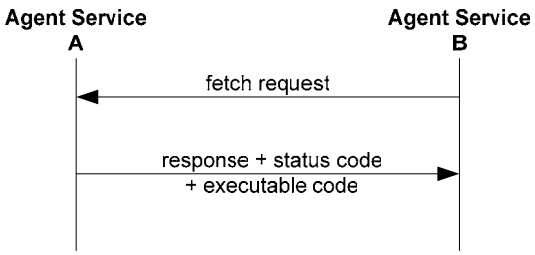
การดึงเอเจนท์กลับจาก B ไป A โดยเอเจนท์จะถูกส่งกลับไปพร้อมกับข้อความตอบรับ การส่งข้อความ retract request แสดงดังภาพประกอบ 2.6



ภาพประกอบ 2.6 การส่งข้อความ retract request

3. Fetch the class

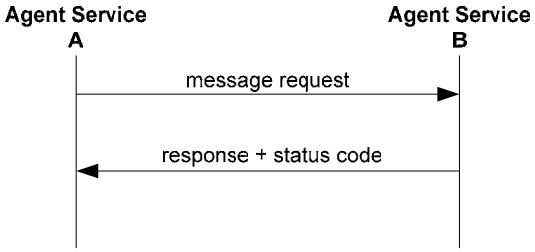
ในการประมวลผลเอเจนท์นั้นเอเจนท์เซอร์วิส B ต้องร้องขอรหัสคำสั่งสำหรับการประมวลผลของเอเจนท์จากเอเจนท์เซอร์วิส A ซึ่งจะถูกส่งมาพร้อมกับข้อความตอบรับ การส่งข้อความ fetch request แสดงดังภาพประกอบ 2.7



ภาพประกอบ 2.7 การส่งข้อความ fetch request

4. Send a message:

การส่งข้อความจาก A ไป B โดยข้อความจะถูกส่งไปพร้อมกับข้อความร้องขอการส่งข้อความ message request แสดงดังภาพประกอบ 2.8



ภาพประกอบ 2.8 การส่งข้อความ message request

2.6 ประเด็นด้านความปลอดภัยในระบบเอเจนท์แบบเคลื่อนที่

แม้ว่าการนำเทคโนโลยีเอเจนท์แบบเคลื่อนที่มาใช้จะช่วยอำนวยความสะดวกในการทำงาน แต่ระบบเอเจนท์แบบเคลื่อนที่ก็ยังมีข้อจำกัดในเรื่องความปลอดภัยซึ่งเป็นสิ่งที่ต้องคำนึงถึงเมื่อมีการพัฒนาระบบ สำหรับหัวข้อนี้จะกล่าวถึงความต้องการของระบบเอเจนท์แบบเคลื่อนที่ที่ปลอดภัยและการคุกคามความปลอดภัยของระบบเอเจนท์แบบเคลื่อนที่

2.6.1 ความต้องการของระบบเอเจนท์แบบเคลื่อนที่ที่ปลอดภัย

ระบบเอเจนท์แบบเคลื่อนที่ที่มีความปลอดภัยต้องมีการจัดการเพื่อให้รองรับกับความต้องการด้านความปลอดภัยต่อไปนี้ (Karnouskos 2000)

1. Confidentiality

ข้อมูลที่ต้องเก็บเป็นความลับที่ส่งไปกับเอเจนท์หรือข้อมูลที่ถูกใช้โดยระบบเอเจนท์ เช่น บันทึกการทำงานเพื่อใช้ในการตรวจสอบระบบ ควรจะยังคงเป็นความลับ การติดต่อทั้งภายในระบบเอเจนท์และระหว่างระบบเอเจนท์ไม่ควรจะถูกเปิดเผยด้วยการเฝ้าติดตามหรือเทคนิคอื่น ๆ

2. Integrity

เอเจนท์ควรจะได้รับการป้องกันจากการแก้ไขรหัสคำสั่ง สถานะ และข้อมูลโดยผู้ที่ไม่มีสิทธิ์ในการแก้ไขหรือโดยไม่ตั้งใจ แต่หากป้องกันไม่ได้ก็จะต้องสามารถตรวจจับได้ว่ามีการเข้าไปแก้ไขเอเจนท์

3. Availability

ระบบเอเจนท์ต้องให้บริการได้ตามที่กำหนดไว้และสามารถเฝ้าติดตามการบริการและการทำงานของเอเจนท์ได้ โดยต้องมีการจัดการทรัพยากร การควบคุมการใช้งานทรัพยากรพร้อมกัน การจัดการการอับจน (deadlock) การตรวจจับและการกู้คืนจากสถานะที่ผิดพลาด เช่น ความผิดพลาดของฮาร์ดแวร์และซอฟต์แวร์เพื่อไม่ให้เกิดการทำงานวนซ้ำไม่สิ้นสุด

4. Non-Repudiation

ระบบเอเจนท์จะต้องการจัดการการห้ามปฏิเสธความรับผิดชอบและมีหลักฐานที่เพียงพอที่จะใช้ในการจัดการกับความขัดแย้งที่เกิดขึ้นได้ เช่น ในกรณีที่เอเจนท์มีการส่งข้อความให้ผู้รับ และผู้รับได้รับการยืนยันว่าใครเป็นผู้ส่ง ทั้งเอเจนท์ที่เป็นผู้ส่งและผู้รับจะไม่สามารถปฏิเสธได้ว่าไม่เกี่ยวข้องกับข้อความนั้นได้ในภายหลัง

5. Accountability

เอเจนท์และระบบเอเจนท์ควรจะตรวจสอบกิจกรรมของตนเองและเก็บข้อมูลไว้เพื่อการแก้ไขข้อผิดพลาดหรือเพื่อประโยชน์ด้านความปลอดภัย การกระทำทุกอย่างในระบบต้องสามารถระบุได้ว่าใครเป็นผู้ดำเนินการและต้องเป็นผู้ที่ได้รับสิทธิ์เท่านั้น รวมทั้งสามารถทำการตรวจสอบได้ในภายหลัง

2.6.2 การคุกคามความปลอดภัยของระบบเอเจนท์แบบเคลื่อนที่

การคุกคามความปลอดภัยที่สำคัญได้แก่ การเปิดเผยข้อมูล การปฏิเสธการบริการ (denial of service) และการแก้ไขข้อมูล โดยอาจเกิดขึ้นในระบบเอเจนท์แบบเคลื่อนที่ดังนี้ (Jansen and Karygiannis 1999)

1. Agent-to-Platform

เป็นการคุกคามที่เอเจนท์จะใช้จุดอ่อนด้านความปลอดภัยของระบบเอเจนท์ในการทำการโจมตีระบบเอเจนท์ ได้แก่

(1) การปลอมตัว (masquerading) โดยเอเจนท์ที่ไม่มีสิทธิ์ใช้งานแพลตฟอร์มทำการปลอมแปลงเอกลักษณ์ (identity) ของตนเป็นเอเจนท์ที่มีสิทธิ์ที่ถูกต้องเพื่อใช้บริการและทรัพยากรของระบบ และใช้สิทธิ์ที่ได้มานั้นทำลายความน่าเชื่อถือของเอเจนท์ที่ถูกต้องได้หรือขโมยข้อมูลที่สำคัญ

(2) การปฏิเสธการบริการ เป็นการคุกคามที่เอเจนท์แบบเคลื่อนที่เข้ามาใช้ทรัพยากรของระบบมากเกินไปจนทำให้ต้องปฏิเสธการให้บริการเอเจนท์อื่น ๆ หรือผู้ใช้ที่มีสิทธิ์

ในการใช้งานของระบบ และอาจทำให้ระบบไม่สามารถให้บริการใดๆ หรือหยุดการทำงานของระบบได้

(3) การเข้าถึงโดยไม่มีสิทธิ์ เนื่องจากเอเจนท์ที่จะเข้าใช้งานระบบจะต้องมีการตรวจสอบสิทธิ์ที่ระบุไว้ในนโยบายความปลอดภัยซึ่งมีกลไกควบคุมการเข้าถึงการบริการและทรัพยากรของระบบ โดยจะต้องมีการให้สิทธิ์แก่เอเจนท์แบบเคลื่อนที่ตามเอกลักษณ์ที่ระบุก่อน หากเอเจนท์ที่เข้าถึงระบบโดยไม่ได้รับสิทธิ์ที่เหมาะสมก็จะเป็นอันตรายต่อระบบได้

2. Agent-to-Agent

เป็นการคุกคามที่เอเจนท์ที่ใช้จุดอ่อนของเอเจนท์อื่นเพื่อทำการโจมตีเอเจนท์นั้น ได้แก่

(1) การปลอมตัว เนื่องจากเอเจนท์สามารถติดต่อกันได้โดยตรงซึ่งอาจมีการพยายามปลอมตัวเป็นเอเจนท์ที่มีสิทธิ์ที่จะดูข้อมูลของเอเจนท์อื่นและให้ข้อมูลที่ไม่ถูกต้องแก่เอเจนท์อื่น

(2) การปฏิเสธการบริการ เอเจนท์สามารถใช้เอเจนท์อื่นเพื่อให้ทำการโจมตีเอเจนท์ด้วยกัน เช่น การส่งข้อความปลอมให้เป็นจำนวนมาก (spamming) เมื่อเอเจนท์ได้รับข้อความแล้วก็ต้องทำการปฏิเสธการรับข้อความนี้ แต่การทำงานนี้ต้องอาศัยทรัพยากรของระบบซึ่งหากเป็นปริมาณมาก ๆ ก็อาจทำให้ระบบให้บริการแก่เอเจนท์อื่นไม่ได้

(3) การปฏิเสธว่าเคยได้รับข้อความ (repudiation) จะเกิดขึ้นเมื่อเอเจนท์ที่มีส่วนในการติดต่อหรือทำรายการข้อมูล (transaction) ปฏิเสธว่าไม่เคยเกิดการติดต่อหรือทำรายการข้อมูลขึ้น ซึ่งระบบจะต้องมีหลักฐานเพียงพอที่จะใช้ในการจัดการกับการขัดแย้งที่เกิดขึ้นได้

(4) การเข้าถึงโดยไม่มีสิทธิ์ หากระบบเอเจนท์มีจุดอ่อนหรือไม่มีกลไกในการควบคุม เอเจนท์ที่ประสงค์ร้ายอาจรบกวนเอเจนท์อื่นได้โดยตรงด้วยการเรียกส่วนการทำงานที่เป็นสาธารณะ (public) เพื่อทำให้เกิดการใช้บัฟเฟอร์เกินขนาด (buffer overflow) หรือการแก้ไขรหัสคำสั่งหรือข้อมูลของเอเจนท์อื่น หรือการเฝ้าติดตามกิจกรรมของเอเจนท์อื่นโดยใช้บริการของระบบเอเจนท์เพื่อคอยดักฟังการติดต่อสื่อสารได้

3. Platform-to-Agent

เป็นการคุกคามที่ระบบเอเจนท์ทำให้เกิดความไม่ปลอดภัยแก่เอเจนท์ ได้แก่

(1) การปลอมตัว เช่น ระบบเอเจนท์ปลอมตัวเป็นระบบเอเจนท์ปลายทางหรือเป็นระบบเอเจนท์ที่ไว้ใจได้เพื่อให้เอเจนท์เปิดเผยข้อมูลสำคัญ หรือทำการแก้ไขรหัสคำสั่ง ข้อมูลสถานะของเอเจนท์ได้

(2) การปฏิเสธการบริการ เมื่อเอเจนท์มาถึงระบบเอเจนท์ปลายทางเพื่อทำการประมวลผล แต่ระบบเอเจนท์ปลายทางซึ่งประสงค์ร้ายอาจปฏิเสธการร้องขอการใช้ทรัพยากรไม่ให้เอเจนท์ประมวลผล หรือหยุดการทำงานของเอเจนท์ ซึ่งจะทำให้ระบบเอเจนท์อื่นที่รอผลลัพธ์ของเอเจนท์อาจเกิดการอับจนได้

(3) การดักฟัง (eavesdropping) ระบบเอเจนท์อาจถูกดักฟังหรือถูกเฝ้าติดตามการติดต่อสื่อสารที่เป็นความลับของเอเจนท์ ซึ่งในระบบเอเจนท์แบบเคลื่อนที่นั้นระบบเอเจนท์สามารถติดตามการติดต่อสื่อสารและดูคำสั่งที่เอเจนท์ใช้ประมวลผลทั้งหมด ดังนั้นข้อมูลที่เป็นความลับที่ระบบเอเจนท์มีสิทธิ์หรือข้อมูลที่ไม่ได้เข้ารหัสที่เอเจนท์ต้องนำมาใช้ และผลลัพธ์ที่เกิดจากระบบเอเจนท์จะถูกเปิดเผยแก่ระบบเอเจนท์ ซึ่งระบบเอเจนท์อาจจะนำข้อมูลนี้ไปใช้ร่วมกันกับระบบเอเจนท์อื่นซึ่งอาจจะทำให้เกิดอันตรายแก่เอเจนท์ได้

(4) การแก้ไข (alteration) เมื่อเอเจนท์มาถึงระบบเอเจนท์ก็จะเปิดเผยรหัสคำสั่งสถานะ และข้อมูลแก่ระบบเอเจนท์ ระบบเอเจนท์ที่ประสงค์ร้ายอาจจะทำการแก้ไขรหัสคำสั่งสถานะ ข้อมูลและผลลัพธ์ที่ได้จากระบบเอเจนท์ แต่การแก้ไขรหัสคำสั่งและผลลัพธ์นั้นสามารถตรวจจับได้เพราะมีการลงลายเซ็นต์ดิจิตอลไว้ ดังนั้นระบบเอเจนท์ที่ประสงค์ร้ายก็จะทำการแก้ไขสถานะหรือข้อมูลแทนหรือระบบเอเจนท์อาจจะสร้างเครื่องเสมือน (virtual machine) ปลอมเพื่อสร้างผลลัพธ์ที่ไม่ถูกต้องได้

4. Other-to-Agent Platform

การคุกคามต่อระบบเอเจนท์ ได้แก่

(1) การปลอมตัว เอเจนท์อาจปลอมตัวเพื่อร้องขอบริการหรือทรัพยากรจากระบบเอเจนท์ที่ตนไม่มีสิทธิ์

(2) การเข้าถึงโดยไม่มีสิทธิ์ เนื่องจากผู้ใช้ โปรเซส และเอเจนท์ที่อยู่ระยะไกลอาจร้องขอทรัพยากรที่ตนไม่มีสิทธิ์ได้ ดังนั้นควรมีการป้องกันการเข้าถึงระบบเอเจนท์ในระยะไกลเพราะอาจทำให้ถูกโจมตีได้ หรือถูกทำให้ระบบล่ม หรือถูกควบคุมทรัพยากรทั้งหมดของระบบได้

(3) การปฏิเสธการบริการ เนื่องจากบริการของระบบเอเจนท์นั้นให้บริการได้ทั้งแบบระยะไกลและภายในเครื่องเดียวกัน ซึ่งการให้บริการเอเจนท์นั้นเป็นการติดต่อระหว่างระบบเอเจนท์ที่อาจจะทำให้เกิดการโจมตีเพื่อให้เกิดการปฏิเสธการให้บริการได้

(4) การทำสำเนา (copy) และทำซ้ำ (replay) ทุกครั้งที่เอเจนท์ย้ายจากระบบเอเจนท์หนึ่งไปอีกระบบเอเจนท์หนึ่งก็ยิ่งเพิ่มโอกาสที่จะถูกคุกคามความปลอดภัย อาจมีการดักจับข้อความของเอเจนท์ในระหว่างเดินทาง ทำสำเนาเอเจนท์หรือข้อความของเอเจนท์ โดยทำซ้ำไว้และส่งเอเจนท์นี้ซ้ำ เป็นต้น

2.7 เอเจนท์แบบเคลื่อนที่กับตลาดอิเล็กทรอนิกส์

การพาณิชย์อิเล็กทรอนิกส์ได้เข้ามามีบทบาทต่อการทำธุรกรรมบนเครือข่ายอินเทอร์เน็ตในปัจจุบันเป็นอย่างมาก เนื่องจากเป็นช่องทางหนึ่งที่จะทำให้ผู้ซื้อและผู้ขายสามารถติดต่อทำธุรกรรมผ่านตลาดอิเล็กทรอนิกส์ ซึ่งเป็นสถานที่ในเครือข่ายที่เกิดการปฏิสัมพันธ์และเกิดการแลกเปลี่ยนข้อมูล สินค้า บริการ และการจ่ายเงิน (Turban et. al 2000) ซึ่งการเปลี่ยนรูปแบบของการซื้อขายมาเป็นการซื้อขายผ่านตลาดอิเล็กทรอนิกส์นั้นช่วยให้เกิดความสะดวกรวดเร็วและลดค่าใช้จ่ายให้แก่องค์กรธุรกิจและลูกค้าทั้งในเรื่องการติดต่อซื้อขาย การขนส่ง และการชำระเงิน ซึ่งจากแนวคิดที่ว่าเอเจนท์แบบเคลื่อนที่เป็นตัวแทนขององค์กรหรือผู้ที่เป็นเจ้าของเอเจนท์ ดังนั้นเมื่อนำเอเจนท์แบบเคลื่อนที่ไปใช้ในตลาดอิเล็กทรอนิกส์ก็เสมือนกับว่าผู้ที่เป็นเจ้าของเอเจนท์นั้นเป็นผู้ทำธุรกรรมในตลาดอิเล็กทรอนิกส์ ตัวอย่างการนำเอเจนท์แบบเคลื่อนที่ไปประยุกต์ใช้ในตลาดอิเล็กทรอนิกส์ (He and Leung 2002) เช่น การใช้เอเจนท์ในการติดตามข้อมูล วิเคราะห์ข้อมูลหรือการทำรายการข้อมูล ต่าง ๆ แทนผู้ที่เป็นเจ้าของเอเจนท์ การใช้เอเจนท์เพื่อหาแนวทางที่ดีที่สุดสำหรับลูกค้า เช่น บอกได้ว่าวิธีไหนที่ทำให้เกิดค่าใช้จ่ายต่ำสุด หรือตอบสนองความต้องการได้เร็วกว่า การใช้เอเจนท์ในการตัดสินใจหรือการต่อรองราคาในการทำธุรกรรม เป็นต้น

2.7.1 การประมูลอิเล็กทรอนิกส์

การประมูลอิเล็กทรอนิกส์เป็นรูปแบบหนึ่งของตลาดอิเล็กทรอนิกส์ที่ได้รับความนิยมอย่างมากในปัจจุบัน การประมูลจะทำผ่านเซิร์ฟเวอร์ที่ให้บริการบนอินเทอร์เน็ตซึ่งผู้จัดการประมูลจะทำหน้าที่เสมือนนายหน้าเสนอสินค้าและบริการของผู้ที่ต้องการขายพร้อมทั้งรายละเอียดสินค้าและบริการ ผู้ประมูลทำการเสนอราคาประมูลผ่านแบบฟอร์มอิเล็กทรอนิกส์โดยชื่อผู้ที่เสนอราคาจะถูกเก็บเป็นความลับ ส่วนค่าใช้จ่ายหรือค่าธรรมเนียมนั้นก็เป็นไปตามกฎที่ผู้จัดประมูลกำหนด (Turban et. al 2000) การประมูลอิเล็กทรอนิกส์ช่วยลดข้อจำกัดของการประมูลปกติโดยผู้ประมูลไม่จำเป็นต้องเดินทางมายังสถานที่ในการประมูลด้วยตนเองแต่จะทำการเสนอราคาประมูลผ่านทางเครือข่ายอินเทอร์เน็ตซึ่งทำได้สะดวกรวดเร็ว โดยทั่วไปแล้ววิธีการประมูลมีหลายรูปแบบ (Garcia et. al 2001) เช่น

- **English auction** เป็นการประมูลโดยกำหนดราคาขั้นต่ำไว้ ผู้ที่เข้าร่วมประมูลจะเสนอราคาสูงขึ้นเรื่อยๆ จนกว่าจะไม่มีผู้ใดให้ราคาสูงกว่า หรือตรงตามเงื่อนไขของเจ้าของสินค้า ผู้ที่ให้ราคาสูงสุดจะได้รับสินค้าไป

- **Dutch auction** เป็นการประมูลโดยทำการตั้งราคาสูงสุดไว้ ผู้เข้าร่วมประมูลจะเสนอราคาต่ำลงเรื่อยๆ จนกว่าจะมีผู้ที่พอใจราคานั้น
- **American auction หรือ Closed bid auction** เป็นการประมูลที่การเสนอราคาเป็นความลับ โดยเมื่อถึงเวลาเปิดซองประมูล ผู้ที่ให้ราคาสูงสุดจะเป็นผู้ชนะการประมูล
- **Vickrey Auction หรือ Uniform second auction** เป็นการประมูลที่การเสนอราคาเป็นความลับเช่นเดียวกับ American Auction และผู้ให้ราคาสูงสุดเป็นผู้ชนะแต่จะจ่ายเงินในราคาเท่ากับที่ผู้เสนอเป็นอันดับสองเท่านั้น

2.7.2 ระบบเอเจนท์แบบเคลื่อนที่สำหรับการประมูลอิเล็กทรอนิกส์

ตัวอย่างของระบบการประมูลที่นำเทคโนโลยีเอเจนท์มาใช้ในตลาดอิเล็กทรอนิกส์ได้แก่ Nomad และ MoCAAS

2.7.2.1 Nomad

Nomad (Sandholm and Huai 2000) เป็นระบบเอเจนท์แบบเคลื่อนที่ที่รวมอยู่ในเครื่องแม่ข่ายสำหรับการประมูลผ่านอินเทอร์เน็ตที่เรียกว่า eAuctionHouse ซึ่งเป็นตลาดอิเล็กทรอนิกส์ให้ผู้ซื้อและผู้ขายสามารถซื้อขายสินค้า เสนอราคาประมูล เริ่มเปิดการประมูลและปิดการประมูลเองได้

เอเจนท์ใน Nomad มี 2 แบบคือ

(1) Tailored agent เป็นเอเจนท์ที่ผู้ใช้เขียนโปรแกรมเอเจนท์ขึ้นเองด้วยภาษาจาวา

(2) Template agent เป็นเอเจนท์ที่สร้างจากเอเจนท์แม่แบบของ Nomad ที่มีอยู่แล้ว เช่น แม่แบบสำหรับสร้างเอเจนท์ที่ทำหน้าที่รวบรวมข้อมูล ติดตามการประมูลและแจ้งข่าวสารให้แก่ผู้ใช้ผ่านทางจดหมายอิเล็กทรอนิกส์ แม่แบบสำหรับสร้างเอเจนท์ที่ทำหน้าที่ประมูลแบบ English auction ซึ่งจะเสนอราคาต่ำสุดที่มากกว่าราคาที่เสนอสูงสุดในขณะนั้นและจะหยุดเสนอราคาเมื่อถึงราคาที่ผู้ใช้กำหนดไว้ รวมทั้งแม่แบบสำหรับการประมูลแบบอื่น ๆ

องค์ประกอบของ Nomad มี 4 ส่วนหลักคือ

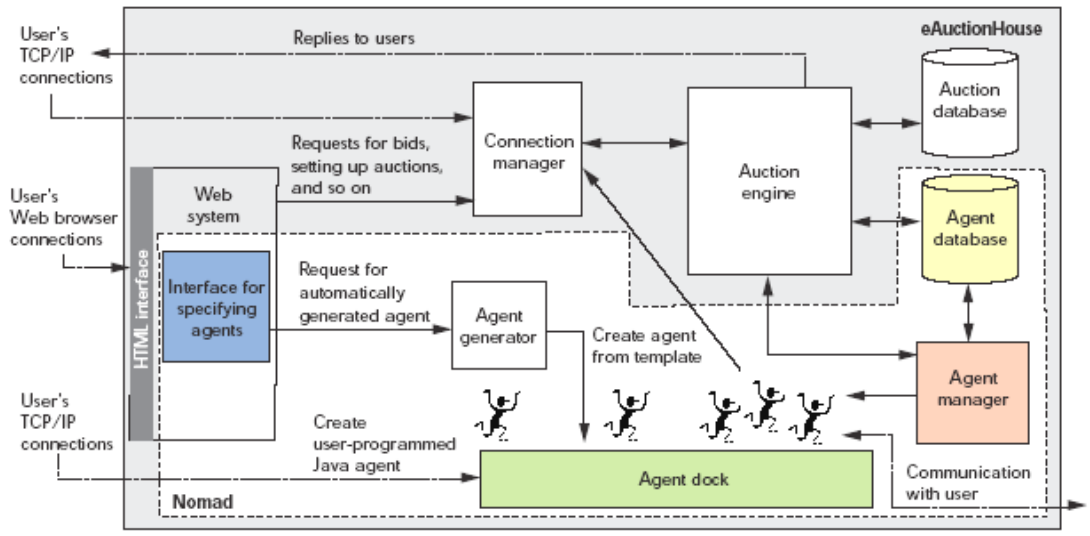
(1) Interface for specifying agents เป็นส่วนติดต่อผู้ใช้สำหรับระบุชนิดของเอเจนท์ที่ต้องการสร้าง

(2) Agent dock เป็นสภาวะแวดล้อมในการประมวลผลของเอเจนท์

(3) Agent manager เป็นส่วนควบคุมการจัดการเอเจนท์

(4) Agent database เป็นฐานข้อมูลเอเจนท์

สถาปัตยกรรมของ Nomad ภายใน eAuctionHouse แสดงดังภาพประกอบ 2.9



ภาพประกอบ 2.9 สถาปัตยกรรมของ Nomad ภายใน eAuctionHouse (Sandholm and Huai 2000)

การสร้างเอเจนท์แม่แบบเอเจนท์จะทำผ่านส่วนติดต่อผู้ใช้ในระบบเว็บโดยคำสั่งในการร้องขอจะถูกส่งไปยัง Agent generator ส่วน เอเจนท์ที่ผู้ใช้สร้างขึ้นเองจะสร้างผ่านเครือข่าย โดยเอเจนท์ที่สร้างขึ้นจะถูกลงทะเบียนที่ Agent dock ด้วยข้อมูลของผู้ที่สร้างเอเจนท์ขึ้นซึ่งมี Agent manager จัดเก็บข้อมูลเกี่ยวกับเอเจนท์ไว้ใน Agent database และจัดการแจ้งข้อมูลเกี่ยวกับการประมูลให้แก่เอเจนท์เมื่อมีการเปลี่ยนแปลงที่เกี่ยวกับการประมูลเกิดขึ้นแก่ผู้ใช้ทางจดหมายอิเล็กทรอนิกส์

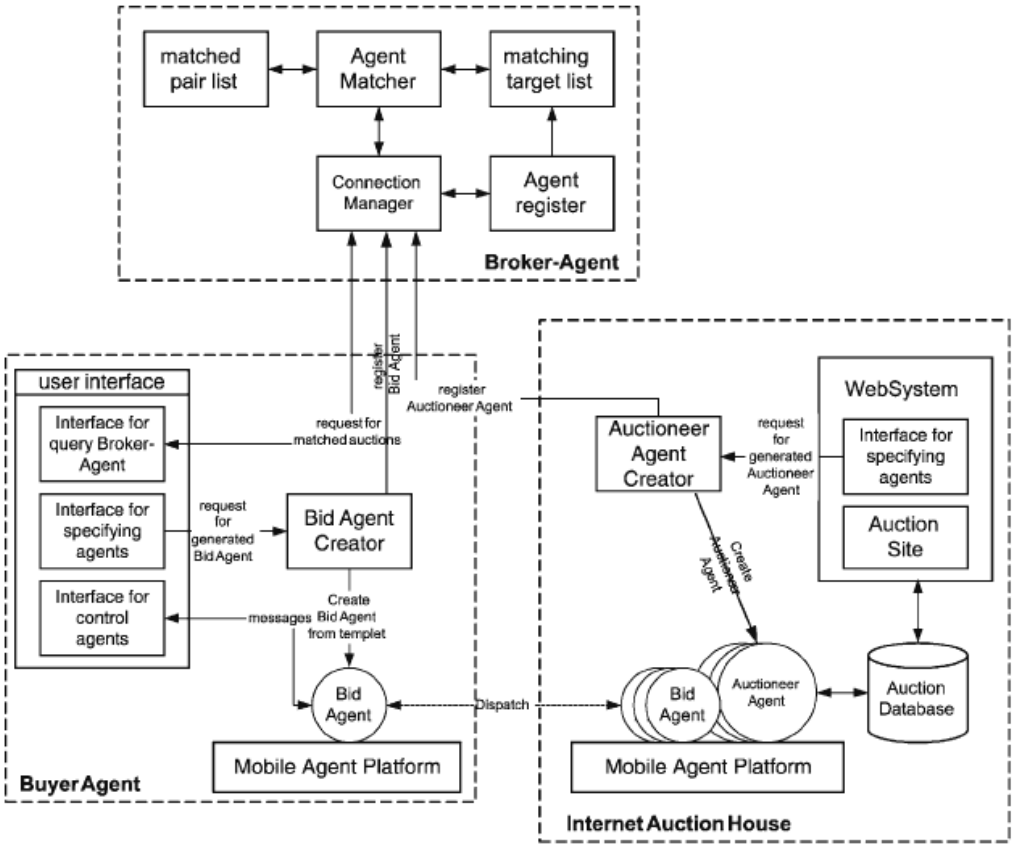
เมื่อเอเจนท์เริ่มประมวลผลใน Agent dock ก็จะเริ่มทำงานตามหน้าที่ที่ถูกกำหนดไว้ หากเป็นเอเจนท์ที่สร้างจากแม่แบบ Information agent ก็จะทำการรวบรวมข้อมูลที่เกี่ยวกับการประมูล หรือหากเป็นเอเจนท์ที่สร้างจาก Incrementor agent ก็จะทำการประมูลตามเงื่อนไขที่กำหนดไว้

คำสั่งในการเริ่มการประมูลและการเสนอราคาใน eAuctionHouse จะถูกส่งจากผู้ใช้หรือจากเอเจนท์ไปยัง Connection manager เพื่อทำการตรวจสอบข้อมูลที่ร้องขอและหากการร้องขอตรงกับเงื่อนไขที่กำหนดใน Auction engine เช่น เป็นการเสนอราคาประมูลที่ตรงกับเงื่อนไขก็จะดำเนินการปรับปรุงข้อมูลการประมูลใน Auction database

โดยสรุปแล้ว Nomad เป็นระบบเอเจนท์แบบเคลื่อนที่ที่รวมอยู่ภายใน eAuctionHouse เพื่อสร้างเอเจนท์สำหรับเข้าร่วมประมูล มีข้อดีคือ มีแม่แบบสำหรับการสร้างเอเจนท์สำหรับการประมูลแบบต่าง ๆ ไว้ให้ ผู้ประมูลต้องขอข้อมูลสำหรับการประมูลจาก eAuctionhouse และเอเจนท์ของ Nomad สามารถเข้าร่วมประมูลได้แม้ว่าผู้ใช้จะไม่ได้เชื่อมต่อกับเครือข่ายในระหว่างที่มีการประมูลโดยข้อมูลที่เกี่ยวข้องกับการประมูลจะถูกส่งให้แก่ผู้ประมูลผ่านทางจดหมายอิเล็กทรอนิกส์

2.7.2.2 MoCAAS

MoCAAS (Lee et. al 2003) เป็นระบบเอเจนท์สำหรับการประมูลโดยใช้เอเจนท์แบบเคลื่อนที่ทำงานร่วมกันในการประมูลแบบ English auction ประกอบด้วยเอเจนท์ของผู้ซื้อ เอเจนท์ของผู้ขาย และเอเจนท์ตัวกลางระหว่างผู้ซื้อและผู้ขาย สถาปัตยกรรมของ MoCAAS แสดงดังภาพประกอบ 2.10



ภาพประกอบ 2.10 สถาปัตยกรรมของ MoCAAS (Lee et. al 2003)

สถาปัตยกรรมของ MoCAAS แบ่งเป็น 3 ส่วนหลักคือ

(1) Buyer-agent ประกอบด้วย

- ส่วนติดต่อผู้ใช้สำหรับการสอบถาม Broker-agent เพื่อแสดงรายชื่อของผู้เสนอขายสินค้า (auctioneer list) และราคาเป้าหมายที่คาดหวังของสินค้านั้น
- ส่วนติดต่อผู้ใช้สำหรับการกำหนด Bid-agents ซึ่งจะส่งข้อมูลในการสร้างเอเจนท์ให้แก่ Bid-agent creator เช่น วิธีการประมูล กำหนดการประมูล เป็นต้น
- ส่วนติดต่อผู้ใช้สำหรับการควบคุมและติดต่อกับ Bid-agent
- Bid-agent creator ทำหน้าที่สร้าง Bid-agent จากแม่แบบของ Bid-agent และทำการลงทะเบียน Bid-agent กับ Broker-agent
- Bid-agent ที่ถูกสร้างขึ้นใน Mobile agent platform จะถูกส่งไปยัง Internet auction house

(2) Broker-agent ทำหน้าที่ในการจับคู่ผู้ซื้อและผู้ขายตามราคาที่เสนอซื้อและขาย ประกอบด้วย

- Connection manager ทำหน้าที่ส่งข้อความไปให้ Agent register หรือ Agent matcher โดยหากเป็นข้อความจาก Buyer-agent ในการสอบถามรายการของ Auctioneer-agent แล้ว Connection manager จะส่งข้อความต่อไปให้ Agent matcher แต่ถ้าเป็นข้อความจาก Buyer-agent หรือ Internet auction house เพื่อขอลงทะเบียนเอเจนท์ Connection manager ก็จะส่งข้อความนั้นให้ Agent register
- Agent register ทำหน้าที่ลงทะเบียนเอเจนท์ที่ขอลงทะเบียนไว้ในรายการ matching target list
- Agent matcher ทำหน้าที่จับคู่ Buyer-agent กับ Auctioneer-agent และใส่คู่ของผู้ซื้อและผู้ขายรวมทั้งราคาประมูลไว้ในรายการ matched pair list

(3) Internet auction house เป็นสถานที่ที่มีการประมูลเกิดขึ้น ประกอบด้วย

- ส่วนติดต่อผู้ใช้สำหรับกำหนดการสร้าง Auctioneer-agent และส่งการร้องขอไปยัง Auctioneer-agent creator
- Auctioneer-agent creator ทำการสร้าง Auctioneer-agent จากแม่แบบและลงทะเบียน Auctioneer-agent ไว้กับ Broker-agent
- Auctioneer-agent ที่ถูกสร้างขึ้นจะรอ Bid-agent ใน Mobile agent platform

ในกระบวนการประมูลนั้น เมื่อ Bid-agent เดินทางไปยัง Internet auction house ก็จะลงทะเบียนกับ Auctioneer-agent หากข้อมูลของ Bid-agent ถูกต้องก็จะตรวจสอบราคาประมูล ซึ่งหากถูกต้องก็จะรับการเสนอราคาประมูลและเก็บข้อมูลไว้ เมื่อสิ้นสุดการประมูล Auctioneer-agent จะแจ้งผลการประมูลให้แก่ Bid-agent ทั้งหมดที่ลงทะเบียนไว้กับ Auctioneer-agent และทำรายการข้อมูลกับ Bid-agent ที่ชนะการประมูล

โดยสรุปแล้ว MoCAAS เป็นระบบเอเจนท์แบบเคลื่อนที่สำหรับการประมูลแบบ English auction โดยมีเอเจนท์สำหรับผู้ซื้อ ผู้ขาย และเอเจนท์ที่ทำหน้าที่เป็นตัวกลางทำงานร่วมกันในการประมูลและจับคู่ผู้ซื้อและผู้ขายที่เหมาะสม

2.8 สรุป

เทคโนโลยีเอเจนท์แบบเคลื่อนที่เป็นเทคโนโลยีที่สามารถนำมาประยุกต์กับระบบงานที่มีการใช้งานผ่านเครือข่ายซึ่งเป็นลักษณะการประมวลผลแบบกระจาย การพาณิชย์อิเล็กทรอนิกส์เป็นตัวอย่างงานประยุกต์รูปแบบหนึ่ง ทั้งนี้ได้มีการกำหนดมาตรฐานสำหรับเทคโนโลยีเอเจนท์เพื่อให้ระบบเอเจนท์ต่างระบบกันสามารถทำงานร่วมกันได้ แต่อย่างไรก็ตามประเด็นด้านความปลอดภัยก็ยังเป็นสิ่งที่ต้องคำนึงถึงในการนำระบบเอเจนท์แบบเคลื่อนที่ไปใช้ในการพัฒนาระบบงานประยุกต์